Case Study 2
Database Design & Prototype

Introduction

During the case study week, you should work through this document. In the process of this case study you will

- analyse a case study and make appropriate design decisions
- build an ERD using an appropriate modelling tool
- build a database on the Infotech server with appropriate data
- create queries that would be suitable for determining data to be used in reports.

Study time and materials

Suggested study time

Study learning materials	4.0 hrs
Undertake activities and review activities	6.0 hrs
Total:	*10.0 hrs*

Materials required

Access to the Internet and a web browser for phpMyAdmin;

Access to either Microsoft Visio; alternate modelling tool such as Dia or Lucidchart; or Microsoft Word. While it is possible to create these models in Word, it is recommended that you use Visio or an alternative modelling program as this will provide you with tools and format templates more aligned to the tasks.

Files in "Case Study 2" folder on MySCU under Modules.

Scenario – General information

Rainbow Research is a small research company situated in Byron Bay, NSW Australia. Their clients are varied and they find that the projects they get often have very different requirements. They have a team of researchers who collect the data which is later analysed for the client.

The Solstice Sounds Festival, held in the region, has asked Rainbow Research (RR) to do some research about the people who attend their festivals. The festival's security company has provided them with a list of characteristics, including suspicious behaviours, which RR has been asked to observe and record at the next festival. RR are keen to automate the data collection process and will be providing tablets with a data collection application (RR App) to the researchers.

The festival is implementing a new security system which will allow them to record accurate data about the movements of the festival-goers. The RR App allows the *researchers* to record their observations about festivalgoers. It also allows them to keep in touch with management and security via their unique email address and their mobile for instant messages. .

***People* Observations (Initial Entry to Festival)**: When a person arrives at the festival and their ticket is validated, a festival wristband is attached to their wrist. Researchers will record the characteristics for each of the festivalgoers as they first enter the festival. The RR App will automatically record the id of the researcher and the date/time it was recorded. They want to record their gender, their estimated age group (<20, 20-30, 31-40, 41-50, 51-60, > 60) and what they were wearing (Hoody? Jeans? Thongs? Dress?). The festival security firm also wants information specifically about the people who are wearing a hoody when they enter the festival. If they are wearing a hoody, the observers must record the size of the group they are part of, whether their hoody is up, whether they have bulging pockets, a suspicion rating (from 1 to 9) and any relevant notes. If they aren't wearing a hoody, the observers record if they are wearing a hat and hair colour.

Gate Movement: Every time a person goes in or out of the festival gates, their wristband is scanned. The new security system automatically validates their wristband and records the wristband id, the date time, the gate code and whether they went in or out of the festival area. The new security system is separate and you don't have to consider the wristband validation process. There are 3 gates (Main, South, East).

Observations: The festival area is broken up into ***zones***, each having a zone id (e.g M1, M2, M3, C1, etc.) and a full name (e.g. Mojo Tent, Mojo North, Mojo South, Crossroads, Jambalaya, Techno, Food1, Food2, etc). The size of each zone is recorded in hectares (e.g. 0.75, 1.2) for use in later analysis. The researchers work within their allocated zone and record their ***observations,*** on the RR App, about the count of many people are observed dancing, eating & drinking.

Hoody Activities: As part of their observations in the RR App, the researchers also record the activities of people wearing hoodies. In the RR App, the date and time is added automatically and the researcher selects the activity types from a drop-down. RR wants the ability to easily add or change these activities. In the current brief, the security company wants to record the activity types (standing near the toilet, approaching strangers, crazy behaviour and falling over) and a Suspicion Rating (1-9) needs to be set for each activity type and used for later analysis. Sometimes, the researcher can scan a hoody-wearer's wristband and record this, but mostly the activities are recorded without the wristband being identified.

Your task is to design and develop a database that will support the RR App requirements. The database needs to store both the wristband gate in/out information as well as all observations made by the researchers. You are required to develop the EERD, database prototype and test queries.

Solution Step 1: Design Logical Model

Activity CS2.1 Identify the entities

First we need to determine the entities involved in the system. Remember that we only want to include entities about which we wish to record information.

Write down the entities in the system, and then check your answer with the Feedback section.

_____

_____

_____

Activity CS2.2 Identify the attributes

Identify the attributes that will be needed for each entity. Again, we would speak to the stakeholders to find out what information needs to be stored. At this stage we will find that we make some assumptions. Write these down as you go.

Use the templates below to write in the name of each entity, and then the attributes.

Assumptions:

Activity CS2.3 Identify the possible primary keys

PEOPLE:

GATE_MOVEMENT:

OBSERVATION:

ACTIVITY:

ZONE:

RESEARCHER:

Activity CS2.4 Resolve any Supertype and Subtypes

Next, we'll refine the design for the PEOPLE entity. Use the templates below to rearrange the attributes of the PEOPLE table, using different entities to record information specific to Hoody and Non-Hoody people.

PEOPLE

HOODY

NON HOODY

What attribute/key would you add to the subtypes to use as the Primary Key?

_____

Which attribute would you use as the discriminator?

_____

Given the fact that a person is either a hoody or a non-hoody, would you say that the category is complete?

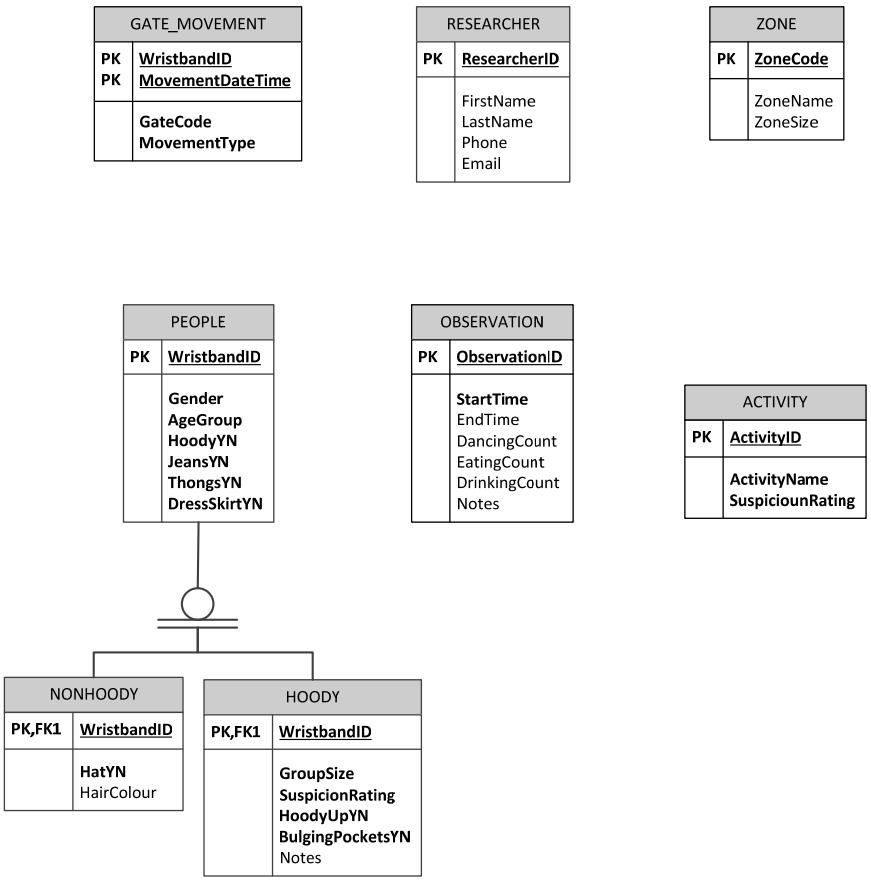
_____

Activity CS2.5 Model the PEOPLE entities

Using Visio or a modelling tool, create a new data model and include all entities and the attributes identified so far. Add each of the people entities you identified above and the super and subtype notation required.

Activity CS2.6 Find relationships between entities

The entities are drawn below. Decide the relationships between the entities (and their cardinality) and draw these on the diagram, writing down any assumptions that you are making.



Assumptions:

_____

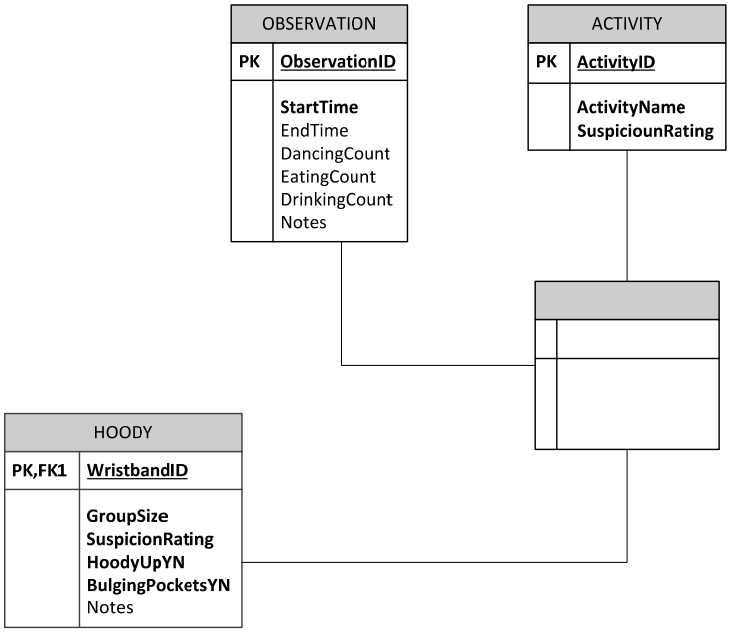
_____

Activity CS2.7 Model the remaining entities

Update your model to include all entities and the attributes identified so far. You do *not* need to add any relationships to this model yet as we will resolve issues first.

Solution Step 2: Normalisation

Activity CS2.8 Resolve many-to-many relationships



Work out the appropriate cardinality for these new relationships, and place on the drawing.

Activity CS2.9 Normalisation

Our relations are now ready to be fully normalised.

PEOPLE (WristbandID, Gender, AgeGroup, HoodyYN, JeansYN, ThongsYN, DressSkirtYN)

GATE_MOVEMENT(WristbandID, MovementDateTime, GateCode, MovementType)

NONHOODY(WristbandID, HatYN, HairColour)

HOODY(WristbandID, GroupSize, SuspicionRating, HoodyUpYN, BulgingPocketsYN, Notes)

RESEARCHER(ResearcherID, FirstName, LastName, Phone, Email)

OBSERVATION(ObservationID, StartTime, EndTime, DancingCount, EatingCount, DrinkingCount)

ZONE(ZoneID, ZoneName, ZoneSize)

ACTIVITY(ActivityID, ActivityName, SuspicionRating)

Normalise each relation, creating new relations if necessary. If no changes are to be made to the above, you can say, 'as above'.

PEOPLE (_____

_____)

GATE_MOVEMENT (_____

_____)

NONHOODY (_____

_____)

HOODY (_____

_____)

RESEARCHER (_____

_____)

OBSERVATION (_____

_____)

ZONE (_____

_____)

ACTIVITY (_____

_____)

(_____

_____)

Solution Step 3: Physical Database Design

Activity CS2.10 Addition of foreign keys

We are now working with the physical model, and need to determine:

- Foreign keys (from the relationships)
- Columns (from the attributes)
- Any constraints (null values, on delete/update constraints).

Identify the foreign keys (if any) that need to be added to the following tables:

PEOPLE _____

GATE_MOVEMENT _____

OBSERVATION _____

HOODY_ACTIVITY _____

Activity CS2.11 Model new entities and add relationships

Update your model to include the new entity and attributes identified so far.

Activity CS2.12 Choice of Data Types and Constraints/Defaults

For each of the following tables, determine the most likely data type for each column, whether it can be NULL, default value if necessary, and whether it is a primary or foreign key or indexed.

PEOPLE						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

GATE_MOVEMENT						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

NONHOODY						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

HOODY						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

RESEARCHER						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

OBSERVATION						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

ZONE						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

ACTIVITY						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

HOODY_ACTIVITY						
Attribute	*Data Type, Size*	*Primary Key*	*Foreign Key*	*Index Type?*	*Default?*	*NULL?*

Solution Step 4: Build the Prototype

Activity CS2.13 Create database and tables in PhpMyAdmin

Create a new database on the Infotech server.

Write the SQL to create the tables in PhpMyAdmin. Do not create the tables until you have added the referential integrity constraints (see next activity).

Activity CS2.14 Create relationships in the database

For each of the following foreign key relationships, decide what should happen when the corresponding key value is updated or deleted.

PEOPLE.WristbandID - > GATEMOVEMENT. WristbandID

PEOPLE.WristbandID - > HOODY.WristbandID

PEOPLE.WristbandID - >.NONHOODY.WristbandID

HOODY.WristbandID - > HOODY_ACTIVITY.WristbandID

OBSERVATION.ResearcherID - > RESEARCHER.ResearcherID

OBSERVATION.ZoneID - > ZONE.ZoneID

HOODY_ACTIVITY.ObservationID - > OBSERVATION.ObservationID

HOODY_ACTIVITY.ActivityID - > OBSERVATION.ActivityID

Write the SQL to create these relationships.

Solution Step 5: Test the Prototype

Activity CS2.15 Create test data in the tables that do not have foreign keys

For each of the following tables, create data that will give meaningful results for various situations.

Activity CS2.16 Create test data in the tables that do have foreign keys

Using the values of the test data already entered, when necessary, fill in some dummy values in the rest of the tables, taking account of referential integrity and possible null values.

Activity CS2.17 Create views

Using the test data, create views which include the following:

1. View to show all information about people with hoodies. The result should show like this:

WristbandID	ResearcherID	Gender	AgeGroup	HoodyYN	JeansYN	ThongsYN	DressSkirtYN	GroupSize	SuspicionRating	HoodyUpYN	BulgingPocketsYN	Notes
1345000123	3	m	<20	1	1	1	0	6	8	1	1	Hmm...
1345007533	3	m	20-30	1	0	0	0	2	0	0	0	*NULL*
1345007833	3	m	20-30	1	1	1	0	3	4	0	0	weird

2. View to show all information about people without hoodies

WristbandID	ResearcherID	Gender	AgeGroup	HoodyYN	JeansYN	ThongsYN	DressSkirtYN	HatYN	HairColour
1345000001	4	m	<20	0	0	0	0	0	red
1345000112	3	m	<20	0	0	0	0	0	light blonde
1345000113	3	f	<20	0	0	0	1	0	dark
1345000114	3	f	20-30	0	0	0	1	0	dark
1345000322	3	f	20-30	0	1	0	0	1	light blonde

3. View to show all information about people with or without hoodies

WristbandID	ResearcherID	Gender	AgeGroup	HoodyYN	JeansYN	ThongsYN	DressSkirtYN	HatYN	HairColour	GroupSize	SuspicionRating	HoodyUpYN	BulgingPocketsYN
1345000001	4	m	<20	0	0	0	0	0	red	NULL	NULL	NULL	NULL
1345000112	3	m	<20	0	0	0	0	0	light blonde	NULL	NULL	NULL	NULL
1345000113	3	f	<20	0	0	0	1	0	dark	NULL	NULL	NULL	NULL
1345000114	3	f	20-30	0	0	0	1	0	dark	NULL	NULL	NULL	NULL
1345000123	3	m	<20	1	1	1	0	NULL	NULL	6	8	1	1
1345000322	3	f	20-30	0	1	0	0	1	light blonde	NULL	NULL	NULL	NULL

4. View to show all information about observations with the researcher name and ZoneName

ObservationID	ResearcherID	ZoneID	StartTime	EndTime	DancingCount	EatingCount	DrinkingCount	ZoneName	Researcher	ZoneSize
3	9	CR03	2013-04-12 13:00:00	2013-04-12 14:00:00	973	120	35	Crossroads South	Jackson Singh	0.30
1	1	CR01	2013-04-12 13:00:00	2013-04-12 14:00:00	1543	126	157	Crossroads Tent 1	Howard Jane	1.00
2	2	CR02	2013-04-12 13:00:00	2013-04-12 14:00:00	1003	55	455	Crossroads Tent 2	Numoto Peter	0.50
4	5	FD01	2013-04-12 13:00:00	2013-04-12 14:00:00	12	3450	2157	Food Area 1	Mason Heather	1.50
5	6	FD02	2013-04-12 13:00:00	2013-04-12 14:00:00	55	2050	1157	Food Area 2	Caruthers Jasmine	1.30
6	7	JM01	2013-04-12 13:00:00	2013-04-12 14:00:00	920	13	87	Jambalaya Tent	Brown Jane	0.40
7	8	TT01	2013-04-12 13:00:00	2013-04-12 14:00:00	3433	10	247	Techno Tent 1	Chan George	1.50

5. View to show all information from HOODY_ACTIVITIES with the ActivityName and SuspicionRating added

HoodyActivityID	ObservationID	ActivityID	WristbandID	Notes	ActivityName	SuspicionRating
1	2	2	1345000123	NULL	standing near the toilet	8
3	2	2	1345000123	NULL	standing near the toilet	8
4	3	3	1345000123	NULL	approaching strangers	8
7	5	3	NULL	NULL	approaching strangers	8
6	3	4	1345000123	NULL	crazy behaviour	7
9	12	4	1345007833	NULL	crazy behaviour	7

Activity CS2.18 Create queries

Using the test data, perform the following queries (sample results shown):

1. Create a query to show the total number of people who attended the festival. Include columns for the total count of the number of people wearing hoodies, jeans, thongs, dresses.

TotalCount	TotalHoody	TotalJeans	TotalThongs	TotalDress
14	3	6	2	5

2. Update the query in step 1 to include columns for all characteristics for the HOODY people.

TotalCount	TotalHoody	TotalJeans	TotalThongs	TotalDress	AvgGroupSize	TotalHoodyUp	TotalBulgingPockets
3	3	2	2	0	3.6667	1	1

3. Create a query to show the total number of both hoody and non-hoody people who attended. Add a calculated column to show the percentage of each.

TotalCount	TotalHoody	PercentHoody
14	3	21.43

OR

TotalCount	TotalHoody	TotalNonHoody	PercentHoody
14	3	11	21.43

4. Create a query to show the count of all hoody and non-hoody characteristics, using the People_All view created earlier.

TotalCount	TotalHoody	TotalJeans	TotalThongs	TotalDress	sum(HatYN)	avg(GroupSize)	sum(HoodyUpYN)	sum(BulgingPocketsYN)
14	3	6	2	5	2	3.6667	1	1

5. Create a query to show the count of people observed dancing, eating and drinking. Group your results by the Zone they were observed in, include the full zone name

StartTime	ZoneID	ZoneName	count(ObservationID)	sum(DancingCount)	sum(EatingCount)	sum(DrinkingCount)
2013-04-12 13:00:00	CR03	Crossroads South	1	973	120	35
2013-04-12 13:00:00	CR01	Crossroads Tent 1	1	1543	126	157
2013-04-12 13:00:00	CR02	Crossroads Tent 2	1	1003	55	455
2013-04-12 13:00:00	FD01	Food Area 1	1	12	3450	2157
2013-04-12 13:00:00	FD02	Food Area 2	1	55	2050	1157
2013-04-12 13:00:00	JM01	Jambalaya Tent	1	920	13	87
2013-04-12 13:00:00	TT01	Techno Tent 1	1	3433	10	247
2013-04-12 14:00:00	CR03	Crossroads South	1	799	160	225
2013-04-12 14:00:00	CR01	Crossroads Tent 1	1	1113	126	127
2013-04-12 14:00:00	CR02	Crossroads Tent 2	1	1003	55	435

Or using Aliased column headings:

StartTime	ZoneID	ZoneName	DancingCountTotal	EatingCountTotal	DrinkingCountTotal
2013-04-12 13:00:00	CR03	Crossroads South	973	120	35
2013-04-12 13:00:00	CR01	Crossroads Tent 1	1543	126	157
2013-04-12 13:00:00	CR02	Crossroads Tent 2	1003	55	455
2013-04-12 13:00:00	FD01	Food Area 1	12	3450	2157

6. Create a query to show a list of hoody activities, grouped by the ActivityName AND sorted by the SuspicionRating (highest first). Provide a total count for each activity.

SuspicionRating	ActivityName	TotalObserved
9	aggression physical	2
8	approaching strangers	2
8	standing near the toilet	2
7	aggression verbal	1
7	crazy behaviour	2

7. Update the previous query (6) so that it also groups your results by the Zone that they were observed in and include the full zone name and zone size details.

SuspicionRating	ActivityName	ZoneName	ZoneSize	TotalObserved
9	aggression physical	Crossroads South	0.30	1
9	aggression physical	Crossroads Tent 1	1.00	1
8	approaching strangers	Crossroads South	0.30	1
8	approaching strangers	Food Area 2	1.30	1
8	standing near the toilet	Crossroads Tent 2	0.50	2
7	aggression verbal	Crossroads Tent 2	0.50	1
7	crazy behaviour	Crossroads South	0.30	1
7	crazy behaviour	Food Area 2	1.30	1

8. Update the previous query (7) and add a calculated column showing the average number of incidents per hectare.

SuspicionRating	ActivityName	ZoneName	ZoneSize	TotalPerHectare
9	aggression physical	Crossroads South	0.30	3.3333
9	aggression physical	Crossroads Tent 1	1.00	1.0000
8	approaching strangers	Crossroads South	0.30	3.3333
8	approaching strangers	Food Area 2	1.30	0.7692
8	standing near the toilet	Crossroads Tent 2	0.50	4.0000
7	aggression verbal	Crossroads Tent 2	0.50	2.0000
7	crazy behaviour	Crossroads South	0.30	3.3333
7	crazy behaviour	Food Area 2	1.30	0.7692

Solution Step 6: Refine your Design

Activity CS2.19 Problems

Further normalisation:

- The AgeGroup categories (e.g. <20, '20-30', etc.) could have a entity created, allowing it to be used in a drop-down list more easily.
- Valid Hair Colour options could also be created as an entity
- The details of other employees at RR may also need to be recorded. This is so that the RR App can also use stored contact information for all management and other RR staff involved with the festival.
- In the scenario, the test says "The researchers work within their allocated zone and record their observations on the RR App." An allocation table to associate the Researchers to particular zones allow observations to automatically record the zone.
- Additionally, the scenario may not have specifically indicated it, but you would surmise that RR would like to use this app for future data collection at other festivals or events. An Event entity (including the festival details, dates, Location, etc.) could be added and all observations related to it.
- If the RR App is to be secure and ensure that the researcher's collection is specifically theirs, a login will be required. The RESEARCHER (or EMPLOYEE) table would need to be enhanced to include their login and password details.

Discuss how you could you change the design to account for some of these requirements:

_____

_____

_____

_____